Machine Learning for Efficient Neighbor Selection in Unstructured P2P Networks

Robert Beverly[1]   Mike Afergan[2]

[1]MIT CSAIL
rbeverly@csail.mit.edu

[2]Akamai/MIT
afergan@alum.mit.edu

USENIX SysML, 2007

CSAIL

Efficient Neighbor Selection

Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Outline

1 Efficient Neighbor Selection
- Problem Overview
- Neighbor Selection and Self-Reorganization

2 Methodology
- Datasets
- Representing the Dataset
- Learning Task

3 Results
- Training Points
- Prediction Results
- Discussion

Efficient Neighbor Selection
Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Efficient Neighbor Selection
in unstructured P2P networks

Problem Domain

- Unstructured P2P overlays, e.g. Kazaa, Gnutella, etc.

Problem

- Self-reorganization in unstructured P2P overlays promises better performance, scalability and resilience
- But *cost of reorganization may be greater than benefit*!

Neighbor Selection Problem

- Choose neighbors **efficiently** → with few queries
- Choose neighbors **effectively** → with high success

Efficient Neighbor Selection

Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Efficient Neighbor Selection
in unstructured P2P networks

Problem Domain

- Unstructured P2P overlays, e.g. Kazaa, Gnutella, etc.

Problem

- Self-reorganization in unstructured P2P overlays promises better performance, scalability and resilience
- But *cost of reorganization may be greater than benefit*!

Neighbor Selection Problem

- Choose neighbors **efficiently** → with few queries
- Choose neighbors **effectively** → with high success

Efficient Neighbor Selection
Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Efficient Neighbor Selection
in unstructured P2P networks

Problem Domain

- Unstructured P2P overlays, e.g. Kazaa, Gnutella, etc.

Problem

- Self-reorganization in unstructured P2P overlays promises better performance, scalability and resilience
- But *cost of reorganization may be greater than benefit*!

Neighbor Selection Problem

- Choose neighbors **efficiently** → with few queries
- Choose neighbors **effectively** → with high success

Efficient Neighbor Selection
Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Efficient Neighbor Selection
in unstructured P2P networks

Our Approach

- Support Vector Machines (SVMs) and feature selection for classification
- Simulate algorithm using live P2P datasets

Results

- Predict "good" neighbors with over 90% accuracy using minimal knowledge of the neighbor's files or type
- Find neighbors capable of answering *future* queries

Efficient Neighbor Selection
Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Efficient Neighbor Selection
in unstructured P2P networks

Our Approach

- Support Vector Machines (SVMs) and feature selection for classification
- Simulate algorithm using live P2P datasets

Results

- Predict "good" neighbors with over 90% accuracy using minimal knowledge of the neighbor's files or type
- Find neighbors capable of answering *future* queries

Efficient Neighbor Selection

Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Unstructured P2P Networks

- Simple, popular and widely used
- e.g. Gnutella estimated at $\simeq$ 3.5M nodes
- Typically used for file sharing
- Overlay Structure:
  - Organic; nodes interconnect with minimal constraints
  - Nodes are dynamic
- Queries:
  - Flooded through overlay
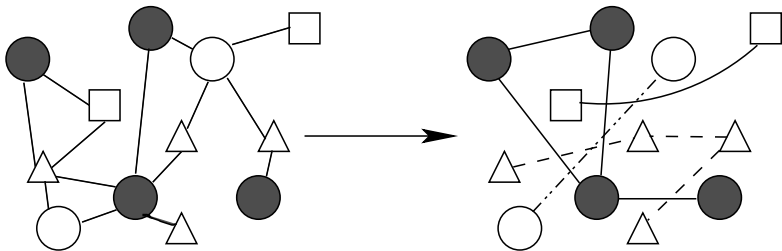 - Peers answer
  - Initiate peer-to-peer download

Efficient Neighbor Selection
Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Outline

1 **Efficient Neighbor Selection**
- Problem Overview
- **Neighbor Selection and Self-Reorganization**

2 Methodology
- Datasets
- Representing the Dataset
- Learning Task

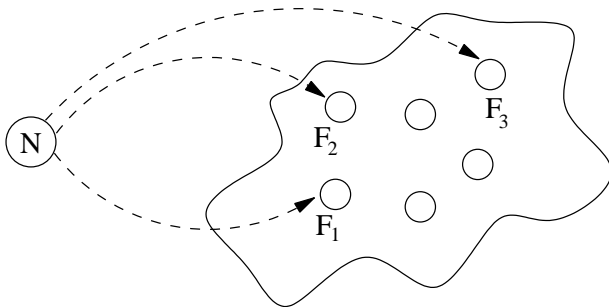**3** Results
- Training Points
- Prediction Results
- Discussion

Efficient Neighbor Selection
Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Self-Reorganization

- Because node connections are unconstrained, previous research suggests *self-reorganization*
- Improved query recall, efficiency, speed, scalability, resilience, trust, etc.

Efficient Neighbor Selection
Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Reorganization Paradox

- But, how can a node determine in real-time whether or not to attach to another node?

Efficient Neighbor Selection

Methodology
Results
Summary

Problem Overview
Neighbor Selection and Self-Reorganization

Reorganization Paradox

- How can a node determine in real-time whether or not to attach to another node?
- Reorganization presents a paradox: only way to learn about another node is to issue queries, but issuing queries reduces the benefit of reorganization.
- Our insight: use machine learning classification plus feature selection

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Outline

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Live P2P Datasets

- Want to evaluate potential algorithms on real data
- Used two Gnutella datasets

DataSet	Nodes	Contains
Beverly, et al.	1,500	Queries, Files, Timestamps
Goh, et al.	4,500	Queries, Files, Timestamps

- Both captured with a promiscuous UltraPeer
- Similar results from both datasets

Efficient Neighbor Selection

Methodology

Results

Summary

Datasets

Representing the Dataset

Learning Task

Outline

CSAIL

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Data Preprocessing

- Nodes hold and advertise files, ex:
    - `"Red Hot Chili Peppers - Californication.mp3"`
- Nodes issue queries, ex:
    - `"remember madonna i'll"` @ 1051761774
- Remove: non-alphanumerics, stop-words, single chars
- Per the Gnutella protocol, we tokenize queries and file name on remaining white space: $\mathbf{f}_i$, $\mathbf{q}_i$
- Let N be the set of all nodes and $n = |N|$.
- Represent all unique tokens and files as $Q = \bigcup \mathbf{q}_i$ and $F = \bigcup \mathbf{f}_i$

Efficient Neighbor Selection
Methodology
Results
Summary

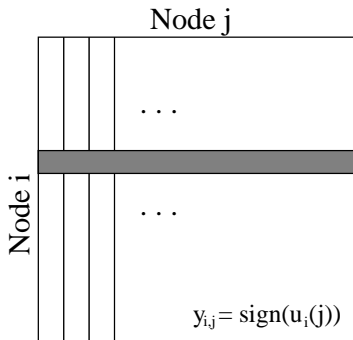Datasets
Representing the Dataset
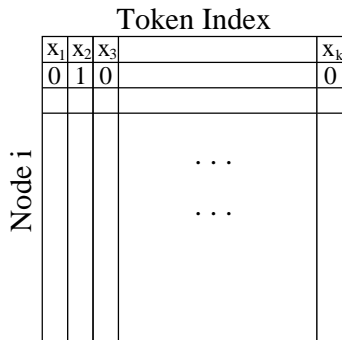Learning Task

Hypothetical Oracle

- Dataset includes all files and queries for every node
- We employ an oracle model in order to measure prediction accuracy
- For every potential connection compute utility $u_i(j)$
- This work defines $u_i(j)$ simply as the number of queries from i matched by j
- Form an n-x-n adjacency matrix **Y** where $Y_{i,j} = \text{sign}\,(u_i((j)))$

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Hypothetical Oracle

Node j

Node i

$y_{i,j} = sign(u_i(j))$

(a) Adjacency Matrix

Token Index

x_1	x_2	x_3		x_k
0	1	0		0

Node i

. . .

. . .

(b) File Store Matrix

Efficient Neighbor Selection
Methodology
Results
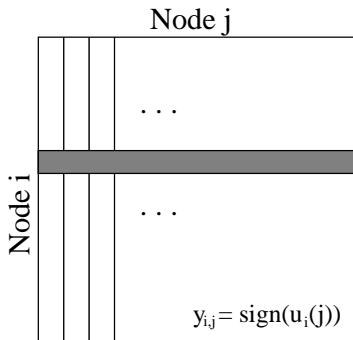Summary

Datasets
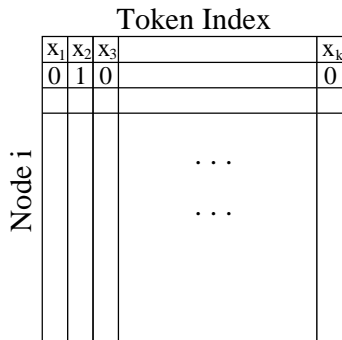Representing the Dataset
Learning Task

Hypothetical Oracle

- Using all file store tokens, F, we assign each token a unique index where $|F| = k$.
- Form an n-x-k file store matrix **X** where $X_{i,j} = 1 \iff F_j \in \mathbf{f}_i$

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Hypothetical Oracle

Node j

Token Index

	x_1	x_2	x_3		x_k
	0	1	0		0

Node i

Node i

. . .

. . .

$y_{i,j} = sign(u_i(j))$

(a) Adjacency Matrix

(b) File Store Matrix

Efficient Neighbor Selection

Methodology

Results

Summary

Datasets

Representing the Dataset

Learning Task

Representing a single node *i*

- The *i*'th row of the adjacency matrix is the first column
- first column represents node *i*'s connection preferences (class labels).
- horizontal concatenation with file store matrix **X**
- Call this oracle representation **Z**

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Outline

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Learning Task

- Given the oracle representation, we turn to ML for classification
- Each node faces a *separate* learning task
- Optimal features will be different for each node and need not match node's queries
    - Node issues queries for "lord of the rings"
    - Best feature: "elves"
    - Intuition: future query for "the two towers" more likely to succeed

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Learning Task
Overview
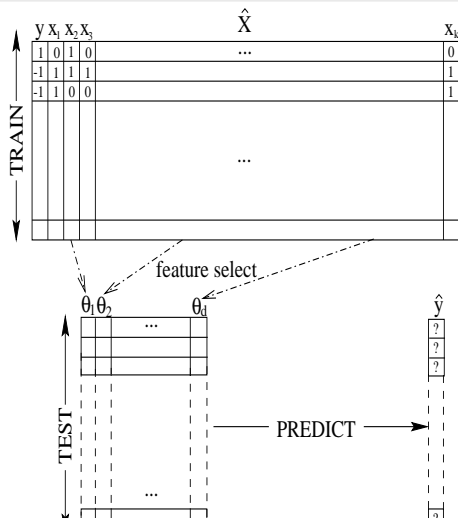
- Randomly permute rows of **Z**
- Select $\ll n$ training nodes
- Learner finds small number $d \ll k$ of features $\theta \in \mathbf{X}$ that best predict y
- Test model on remaining potential peers using θ
- Note features don't contain any queries!

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Key ML Insight for Systems Architects

Feature Selection

Feature selection, variable reduction traditionally used to reduce computational complexity

Key Insight for Systems Architects

Use feature selection to reduce *communication cost*

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Key ML Insight for Systems Architects

Feature Selection

Feature selection, variable reduction traditionally used to reduce computational complexity

Key Insight for Systems Architects

Use feature selection to reduce *communication cost*

Efficient Neighbor Selection
Methodology
Results
Summary

Datasets
Representing the Dataset
Learning Task

Feature Selection

- We consider mutual information (MI) and forward fitting (FF) feature selection
- MI determines how well correlated individual features are to the class label independent of the classifier
- FF greedily finds features that minimize training error for a given classifier

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Outline

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Building a Model
Some questions

Classifier

● Which classifier works best?

Number of Training Points

● Minimum training size that allows for good predictions?

● All results product of five trials with random data permutation

● We find best results with SVMs (also tried Naïve Bayes)

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion
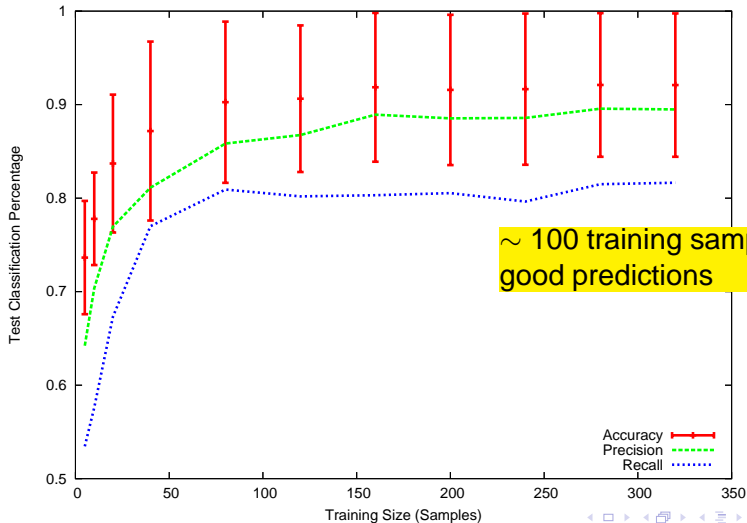
Building a Model
Some questions

Classifier

● Which classifier works best?

Number of Training Points

● Minimum training size that allows for good predictions?

● All results product of five trials with random data permutation

● We find best results with SVMs (also tried Naïve Bayes)

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Building a Model
Some questions

Classifier

- Which classifier works best?

Number of Training Points

- Minimum training size that allows for good predictions?

- All results product of five trials with random data permutation
- We find best results with SVMs (also tried Naïve Bayes)

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Training Points

$\sim$ 100 training samples for good predictions

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Outline

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Prediction Results
Some questions

Number of Features

- How many features are required for accurate predictions?

Feature Selection

- How do FF and MI compare? How much better than random?

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Prediction Results
Some questions

Number of Features
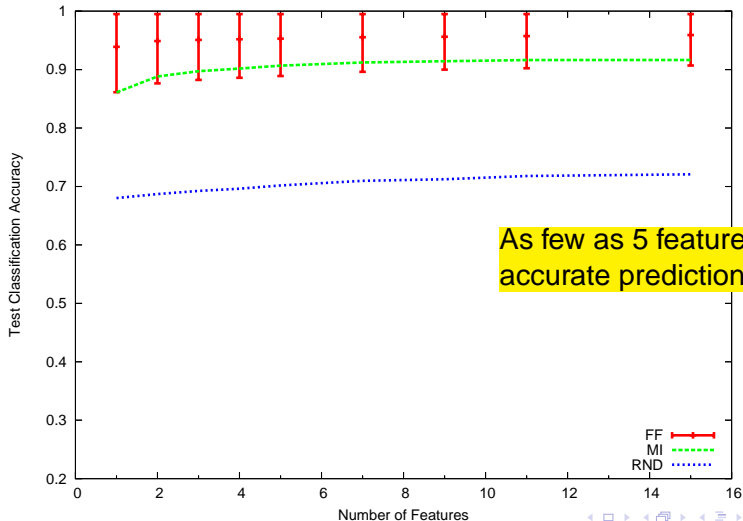
- How many features are required for accurate predictions?

Feature Selection

- How do FF and MI compare? How much better than random?

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Test Accuracy

As few as 5 features give accurate predictions!

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Test Precision

FF outperforms MI

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion
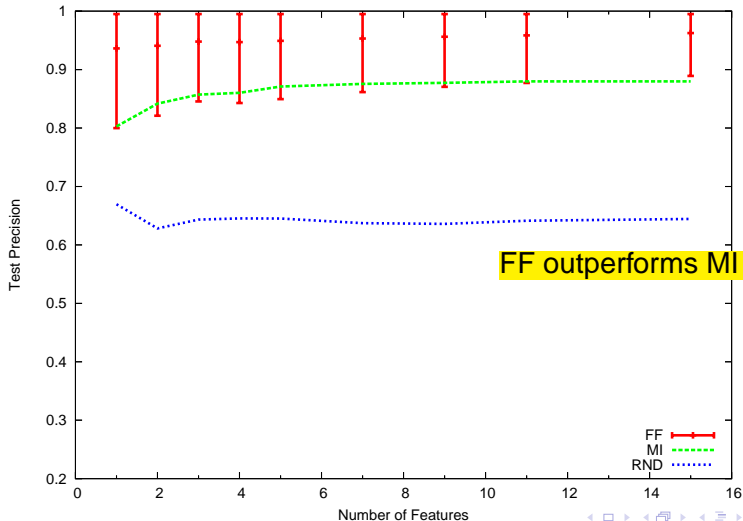
Test Recall
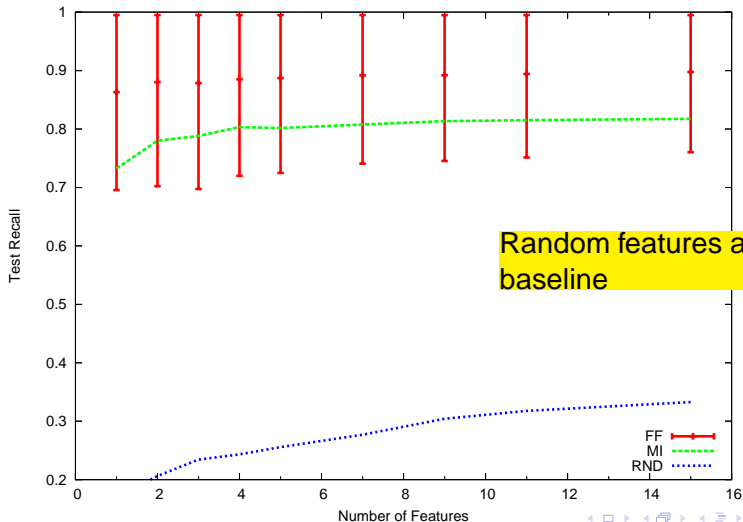
Random features a useful
baseline

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Outline

1. Efficient Neighbor Selection
   - Problem Overview
   - Neighbor Selection and Self-Reorganization

2. Methodology
   - Datasets
   - Representing the Dataset
   - Learning Task

3. Results
   - Training Points
   - Prediction Results
   - **Discussion**

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Discussion

- FF outperforms MI
  - Danger of FF is overfitting, but we do not observe any
  - Consider a node with songs by "Britney Spears"
  - Both "Britney" and "Spears" are good features
  - But with MI, once "Britney" is used, "Spears" doesn't help
  - Future: remove correlations with feature-to-feature MI
- Little SVM overfitting
  - From training size analysis, SVMs are robust to overfitting
  - We do not face the problem of too many features leading to overfitting

Efficient Neighbor Selection
Methodology
Results
Summary

Training Points
Prediction Results
Discussion

Discussion Con't

- Computationally Practical
  - FF requires training a combinatorial number of SVMs
  - Can run as a background process
  - Or, use MI for comparable results
- Practical in real networks
  - Use existing P2P bootstrap mechanisms
  - We show that selecting $\simeq$ 100 nodes suffices to build an effective classifier
- Neighbor Selection is a general problem
  - Feature selection to minimize communication overhead may generalize to other systems/network tasks

Summary

- Novel application of ML to the neighbor selection problem in self-reorganizing networks
- Use feature selection to reduce communication cost in a distributed system
- Correct predictions with over 90% accuracy while requiring minimal queries ($< 2\%$ of features)

Thanks!

Questions?